🎤 Real-Time Audio Translator

CS50 Final Project

Name: [Your Name]
Location: Dublin, Ireland
Date: November 2025
Video Demo: [YouTube Link - To be added]

🌟 Project Overview

The Real-Time Audio Translator is an innovative desktop application that captures audio from any source on your computer, transcribes it using AI, and provides instant translations. Whether you're watching foreign videos, attending international meetings, or learning a new language, this tool breaks down language barriers in real-time.

Key Features

- 🎧 **Universal Audio Capture** - Works with any audio playing on your system
- 🤖 **AI-Powered Transcription** - Uses OpenAI's Whisper for accurate speech recognition
- 🌍 **Multi-Language Support** - Supports 11+ languages with automatic detection
- ⚡ **Real-Time Processing** - See translations as people speak
- 📊 **Smart Voice Detection** - Automatically detects and processes speech segments
- 🎨 **Modern UI** - Clean Windows 11-inspired interface

🛠️ Technology Stack

- **Python 3.8+** - Core programming language
- **Tkinter** - Modern GUI framework
- **Faster-Whisper** - Optimized speech recognition
- **Transformers** - Neural translation models
- **SoundDevice** - Cross-platform audio capture
- **NumPy** - Audio signal processing

📋 Requirements

System Requirements

- Windows 10/11, macOS, or Linux

- 4GB RAM minimum (8GB recommended)
- Python 3.8 or higher
- Audio input device (physical or virtual)

Python Dependencies

```
numpy>=1.21.0
sounddevice>=0.4.6
soundfile>=0.11.0
faster-whisper>=0.10.0
transformers>=4.30.0
torch>=2.0.0
```

🚀 Quick Start

1. **Clone the repository**

   ```bash
   git clone https://github.com/yourusername/real-time-translator.git
   cd real-time-translator
   ```

2. **Install dependencies**

   ```bash
   pip install -r requirements.txt
   ```

3. **Run the application**

   ```bash
   python main_entry.py
   ```

4. **First-time setup**
   - Click "🔄 Refresh" to scan audio devices
   - Select your audio source
   - Click "🤖 Load AI Models" (first load takes 2-3 minutes)
   - Choose source and target languages
   - Click "▶️ Start Translation"

📖 How It Works

Audio Pipeline

1. **Capture** - System audio is captured in real-time chunks
2. **Detection** - Voice Activity Detection identifies speech segments
3. **Buffer** - Speech is buffered until a pause is detected
4. **Process** - Complete segments are sent for transcription

AI Pipeline

1. **Transcription** - Whisper converts speech to text
2. **Language Detection** - Automatic language identification
3. **Translation** - Neural models translate to target language
4. **Display** - Results stream to the UI with natural pacing

🎯 Use Cases

- **Language Learning** - Practice with native content
- **International Meetings** - Real-time meeting translation
- **Media Consumption** - Watch foreign films/videos
- **Accessibility** - Help for hearing impaired users
- **Content Creation** - Generate multilingual subtitles

🔧 Configuration

The application saves your preferences in `config.json`:

```json
{
  "device_id": 1,
  "source_language": "auto",
  "target_language": "es",
  "whisper_model": "base",
  "energy_threshold": 0.01,
  "audio_gain": 1.0
}
```

🐛 Troubleshooting

No Audio Detected

- Ensure correct device is selected
- Run "Test Device" to verify input
- Adjust threshold if too high/low
- Check system audio permissions

Models Won't Load

- Verify internet connection (first download)
- Check available disk space (models need ~1GB)
- Try smaller model size (tiny/base)

Poor Translation Quality

- Use larger Whisper model (small/medium)
- Ensure clear audio source
- Adjust audio gain if needed

🤝 Contributing

Contributions are welcome! Please:

1. Fork the repository
2. Create a feature branch
3. Commit your changes
4. Push to the branch
5. Open a Pull Request

📄 License

This project is licensed under the MIT License - see LICENSE file for details.

🙏 Acknowledgments

- **CS50 Team** - For an amazing course
- **OpenAI** - For the Whisper model
- **Hugging Face** - For translation models
- **Helsinki-NLP** - For OPUS-MT models

📞 Support

For issues, questions, or suggestions:

- Open an issue on GitHub
- Contact: [your-email@example.com]

Made with ❤️ for CS50's Final Project